Before the
Federal Communications Commission
Washington D.C. 20554

In the matter of

Amendment of parts 0, 1, 2, 15, and 18 of the Commission's Rules regarding Authorization of Radiofrequency Equipment	))))	ET Docket No. 15-170
Request for the Allowance of Optional Electronic Labeling for Wireless Devices	))	RM-11673

COMMENT of BRUCE PERENS

To the Commission:

1. Introduction

The rule-making as proposed, and its supporting rationale, do not take into account changes in both the technical framework and the economics of the production of software for wireless devices.

Specifically, Open Source programs such as *Linux*, *BSD*, *Mach*, *OpenWRT*[1], and *Busybox* provide the majority of operating systems and supporting software used in wireless devices today. Open Source underlies the operating system and supporting software used in most WiFi routers, tablets, mobile telephones, and "Internet of Things" (IoT) devices. And yet, the ruling as proposed essentially prohibits the use of Open Source software in the control of the proposed WiFi devices, and in some contexts might prohibit the use of Open Source software as an operating system in systems containing WiFi devices. Given the economic footprint of Open Source in these roles today, the rule-making as currently proposed is a blanket prohibition on an entire well-established industry, and would be likely, if it proceeded without amendment, to incite a court challenge requiring its modification.

The problem before the Commission is the uneasy cohabitation of two parties within the same spectrum: licensed users of weather radar who have priority and who fulfill a mission critical to the safety of life and property, and approximately 800 Million unlicensed WiFi users[2] including a majority of American homes and businesses. In order to manage the conduct of the RF-naïve multitude, the rule-making would require a hardware-enforced lock-down of the software managing WiFi devices, essentially prohibiting Open Source.

This comment proposes to resolve the dichotomy of the need for proper management of WiFi devices and the usual free-for-all of Open Source software. A framework for assuring responsibility for the operation of mass-distributed WiFi control firmware is proposed. This framework gives equal treatment to the original vendors of devices and the Open Source developers.

[1] See http://OpenWRT.org/

[2] Strategy Analytics study, predicting 800 Million WiFi users by the end of 2016, announced in a press release at http://www.businesswire.com/news/home/20120404006331/en/Strategy-Analytics-Quarter-Households-Worldwide-Wireless-Home

Table of Contents

2. Introduction of the Author

The commenter, Bruce Perens, is one of the founders of the *Open Source* movement in software. He is the person who first announced *Open Source* to the world, and the author of *The Open Source Definition,* the rule-set for Open Source licensing. In this work, Mr. Perens is standing on the shoulders of Richard Stallman and his *Free Software* campaign.

Mr. Perens is the creator of *Busybox,* a system management and utility software suite capable of supporting a router or other embedded device. It is a component within many millions of commercial WiFi access points and other WiFi-using devices such as Android phones and tablets, printers, set-top boxes, and IoT devices. Busybox is distributed under an Open Source license.

Mr. Perens is the founder of *No-Code International,* which successfully evangelized the end of the requirement of a Morse code examination as a criterion for Radio Amateur licensing. As a result of his effort and that of many others, the International Telecommunications Union removed *regulation S25.5,* the requirement for a Morse code examination, from the *International Telecommunication Regulations* to which the United States is signatory. This allowed FCC to similarly remove its Morse code requirements in a later rule-making. With the possible exception of Russia, all nations have now discontinued Morse code requirements for Radio Amateurs.

The comments of Mr. Perens are cited in several prior rule-makings, including WT docket 98-143, which reduced the required Morse code speed prior to the ITU change, and RM-11699, which dismisses a proposal to allow increased use of encryption over Amateur Radio.

Mr. Perens has evangelized the development of Open Source codecs for digital voice communications over two-way radio, resulting in the recruitment of a developer who created *Codec2,* an ultra-narrow-band digital voice codec unencumbered by patents or even significant copyright restrictions, and

FreeDV[3], which uses Codec2 to transmit voice over HF radio with greater performance than single-sideband (the best available analog mode) and in half the bandwidth.

Mr. Perens is currently CEO of two companies: *Algoram*[4] is a start-up company producing a handheld software-defined transceiver for 50-1000 MHz, using Open Source software. *Legal Engineering*[5] is the bridge between lawyers and engineers, helping each side to understand the other in situations where law and engineering meet.

Mr. Perens was series editor of the 24-volume *Bruce Perens' Open Source Series*[6] of books published by Prentice Hall PTR. He represented Open Source at the *U.N Summit on the Information Society* at the invitation of the *U.N. Development Program.* He is a professional public speaker and keynotes many legal and technical conferences worldwide. He has been an expert witness or case strategy consultant on seminal court cases, including *Oracle v. Google,* which dealt with Google's right to implement the *Android* system using the *Java* language and API, and *Jacobsen v. Katzer,* which established the legality of Open Source licenses. He has taught Continuing Legal Education programs for attorneys in many states, although he is not an attorney. Mr. Perens holds an Amateur Extra Class operator license, with station license K6BP.

3. A Clash of Paradigms

The operational paradigms of many different groups are in conflict in this proceeding:

- **WiFi users** could have a life-and-property mission or an economic one, or may only be concerned with streaming soap operas. The paradigm of radio is an environment in which users interfere with each other, and one users operation can block others from utilizing the spectrum. Thus, rules must be imposed on all users so that all may share. However, WiFi users are in general unconscious of the existence of other band users and of their own potential to interfere. *An individual users motivation may be to consciously or unconsciously avoid cooperation* and to monopolize the spectrum as necessary for his or her own operations.
- **The Internet designers.** In contrast to radio, Internet users do not interfere with each other, and *all of the incentives are for interoperability.* Thus, the Internet designers have been able to proceed using the paradigm of the Internet Engineering Task Force (IETF), which *doesn't regulate Internet implementations at all.* Those who fail to implement IETF protocols well are punished by the poor performance of their own devices and software. IETF produces technical recommendations exclusively by the consensus of engineers regarding its technical aspects, is deaf to management and many considerations outside of engineering, and has no power to enforce its own recommendations. This group isn't familiar with having to work under a regulatory framework such as that which is absolutely necessary to coordinate use of radio spectrum.
- **Open Source Software Developers** provide the software upon which most WiFi devices are built. Open Source has shown itself to be extremely effective in the production of systems software such as *Linux*

[3] See http://FreeDV.org/
[4] See http://Algoram.com/
[5] See http://LegalEngineering.com/
[6] See http://www.informit.com/promotions/perens-series-page-135563

and *OpenWRT* which together underlie many or most WiFi access points. However, Open Source operates as a loose collaboration between individuals, academia, and business, without management other than consensus among the programmers involved, and there is nothing binding an individual programmer to follow the consensus of others. For Open Source developers the paradigm is to share software in source-code form with the world, and to follow your own individual dream without any control by others.

- **Weather Radar users** have a mission that is critical to life and property, and an economic interest if they commercially report the weather. They are concerned by interference from those 800 Million unlicensed users. Their financial investment in radar infrastructure is significant, often in the Millions of dollars. They prioritize their needs over those of WiFi users, as they have been led to expect from FCC's regulations.
- **WiFi manufacturers** *motivation is to produce a profitable product.* They often leverage Open Source software, using it as the framework of their devices, so that they can focus their development dollars on the business-differentiating aspects of their products rather than infrastructure.
- **FCC** has a mission and requirements imposed by Congress and the Executive Branch, as well as a responsibility to work within international regulations to which the U.S. is signatory, mainly in the form of ITU's *International Telecommunications Regulations.* To fulfill their responsibility, FCC must manage spectrum utilization, prevent interference between users, and protect the licensed users of the radio spectrum from interference originating with unlicensed users. This mission is complicated by the fact that the unlicensed users are not in general conscious of the existence of other band users and are not technically qualified to control their own interference.

Obviously, these communities *will* clash. FCC, the weather radar users, and the WiFi manufacturers are well-educated in the need for regulation. WiFi users are for the most part ignorant of the issues at hand. Internet engineers and the Open Source developers have proceeded with little technical regulation throughout their history and will automatically reject proposals to regulate their activities. And yet some regulation is essential if their systems are to be used by unlicensed sharing partners with licensed users who have priority on the same spectrum. The key is to make such regulation tenable for the Open Source developers and the Internet designers.

4. Embedded Software's Technical Failure and the Need for Open Source Firmware and Operating Systems

The manufacturers who produce WiFi integrated circuits and devices work to a schedule and a budget. Their incentive is to produce software which meets some definition of functionality, ship it, and go on to the next product. In the case of finished WiFi devices, they have little incentive to add previously-unplanned software functionality after a device is sold, even if the hardware is capable of it, reserving such things for their next product as a driver of the sale. They often have little incentive to publish bug fixes as long as their devices meet some minimum functionality, and thus it is normal that all commercial WiFi access points have lacunæ and errata concerning their compliance to the specified protocols. Often they are only tested for their interoperability with a handful of commercial operating systems, rather than their actual conformance with protocols. When protocol testing software or devices are available, the testing coverage is necessarily incomplete. Thus, *all products containing software are* *broken to some extent when purchased, and throughout their existence*. Even if there was such a thing as a perfect implementation, there would be things that we learn about WiFi and Internet protocols after the manufacture of a specific device.

Open Source developers endeavor to repair these issues in their own systems. The *Linux* and *BSD* operating systems, and specialized systems such as the *OpenWRT* WiFi and routing software suite are updated continuously, and are available for a user to install on a computing device even after the original manufacturer has ceased distributing their own updates. Similarly, manufacturers of *Android* devices and other devices containing *Linux* can update their devices as the Open Source software they are based upon improves, without dedicating their own staff to creating that improvement.

It is not unusual that Open Source will be the basis of network research which then propagates to all commercial Internet implementations as a bug fix: For example, Linux was an early implementor of *IPV6,* and was used to test commercial operating systems when they later implemented it. The *Bufferbloat Open Source Project*[7] solved technical issues that severely impaired most home networks and many business ones from utilizing the full performance of their Internet connection. Resolution of these issues allowed real-time telephony and video streaming on networks where that was previously not possible due to inefficient queuing of network packets. The *FQ_CODEL* packet queuing algorithm for governing the flow of network packets, implemented and promoted by the Bufferbloat Project, is a highly-recommended and well-adopted feature for new commercial DSL and cable modems, home and small-office routers and WiFi access points.

By promoting implementations of FQ_CODEL which could be installed on devices which had already been sold, including laptops, tablets, cell phones, and various WiFi and network devices, the Bufferbloat team repaired a severe technical problem of the entire Internet[8].

Open Source is also the basis of very many commercial products and services. It's well documented that giant service businesses like *Google* and *Facebook* are built on a framework of Open Source. Commercial operating systems are as well: the technical basis of MacOS and iOS are Open Source programs including the *Mach* operating system, the *BSD* suite of system utilities, the *KDE* Open Source graphical user interface project, and the *GNU* compiler tools. *Android* is built on top of the *Linux* operating system kernel. After decades of resistance even *Microsoft* has joined Open Source, and today operates many of its own Open Source projects.

5. The Role of Open Source in the Construction of WiFi Devices Today, and How Ill-Considered Regulation Can Harm It

A typical process in the design of the software for a new wireless router is to download the source code of the latest version of the *OpenWRT* Open Source router and access point software, including that

[7] See http://bufferbloat.net/

[8] Of course this repair is still in progress, due to the scale of the Internet and the varying acceptance by manufacturers and system administrators.

project's specific version of *Linux*, and to adapt that software to the needs of the specific product. There are other similar processes for printers, cell phones, and essentially all wireless devices. Many or a majority of these devices start with Open Source.

The rule-making, as written today, would essentially freeze all Open Source WiFi protocol implementations to the versions for current devices, and prevent Open Source developers from developing or testing their code on new WiFi hardware. This would quickly cause the obsolescence of the Open Source code for wireless devices, and would push manufacturers of new products into purchasing other software solutions.

The Open Source developers have historically seen resistance from some integrated circuit manufacturers when they ask for documentation, based on intellectual property considerations. However, the market value of having a device work with *Linux*, *Android*, and other Open-Source-based systems has motivated integrated circuit manufacturers to provide such documentation to Open Source developers, and to make it publicly available on the web in general.

More recently, Open Source developers have experienced resistance from integrated circuit manufacturers in obtaining documentation and development tools. This resistance is attributed to the need of manufacturers to lock down their wireless devices in order to comply with FCC regulation. This has already hindered the use of Linux with new wireless integrated circuits.

Some manufacturers have chosen, in order to conform with recent FCC regulation, to create device drivers for the Linux system for which the source code is trade-secret, in contravention of the copyright licensing conventions used with the Linux system.[9] Such device drivers are likely to be infringing of copyright and in violation of the Digital Millennium Copyright Act. To avoid this conflict, FCC should be careful to avoid introducing regulation compelling locked-down WiFi drivers. That would force companies to infringe copyrights, to avoid implementing their systems for the Linux operating system at all, or it would essentially prohibit the use of the copyright licensing conventions of Linux in a wireless device.

6. The Failure of Modular Implementations of Wireless Devices

Some manufacturers have chosen to implement WiFi devices as separate computers on small modules, where all compliance with FCC regulation is implemented in an embedded processor, and the operating system communicates with the embedded processor to control its operation but does not itself implement the wireless protocols. The operating system developers are not given the capability of modifying the software in the embedded processor.

This paradigm is copied from the design of cellular telephones. However, it is important to note that the cellular networks are licensed systems controlled by their carriers, who have the power to ban a device from their large national systems and can thus compel its implementors to correct software issues,

[9] The GNU General Public License, Version 2, see https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html

including those issues that do not rise to a severity that would require attention from a regulatory authority such as FCC.

There is no such power to compel device manufacturers in connection with WiFi, which in contrast to cellular telephony is constructed of tiny unlicensed networks. WiFi users will not be able to compel a manufacturer to resolve any issue in the modules they use. They can only purchase new hardware and hope it works.

In the case of WiFi, the modular design with a separate embedded processor is a poor engineering practice because it prevents operating system designers from repairing software bugs after the device is manufactured. It brings us back to the failures of embedded software discussed in section 4, above. It casts in concrete buggy software containing lacunæ and errata in its protocol implementations in a module which will be immutable for all time unless the manufacturer can somehow be compelled to update it. Manufacturers can not be expected to continue to update the firmware of such modules, or to pass such updates on in a way that will reach end-user products through the distributors of operating systems. It will not be possible for essential network fixes similar to that for *Bufferbloat* to be added to such modules as problems come up.

7. FCC Regulations Should Not Favor One Intellectual Property Paradigm Over Another

The rule-making as written favors a particular intellectual-property paradigm, *trade-secret*, in which details of WiFi devices are hidden by the manufacturer and parties outside of the manufacturer are not allowed to understand sufficient details of the device necessary to modify it. The imposition of a lock-down requiremet preventing re-flashing or other modification of the software on the device assures this. Lock-down would be enforced either the obscurity of that device's implementation, or through a cryptographic system. In order to keep a system's implementation obscure to the public and thus make it impossible for the public to modify it, it would be necessary to keep implementation details trade-secret. If a cryptographic system is used to lock down the device, more details can be disclosed, but the cryptographic keys would have to be kept secret.

Open Source is the polar opposite of trade-secret, in that all details of Open Source software are disclosed publicly and available for modification without the need to sign anything.

FCC regulation should not make a choice among intellectual property paradigms, as such is outside of their mission and would lead to court challenges. The selection of intellectual property paradigm should be left to the implementor. FCC should enact the minimum regulation necessary to protect licensed users from interference by poorly-modified systems, without favoring any particular intellectual property paradigm.

8. The Requirement for Lock-Down Is Not Justified

The system of locking down firmware specified by the proposed rule-making is based on the assumption that WiFi users will defeat the firmware systems which are used to protect the licensed spectrum partners. These systems detect radar pulses and mark a channel to be protected. The theory behind the current rule-making text seems to be that such defeats will be widespread enough that the conventional means that FCC uses for enforcement: locating the offender through radio-direction finding and individually citing them, will be ineffective.

Modification of WiFi equipment to defeat the facilities that protect radar systems is not widespread today, and does not warrant the lock-down proposed. There has not been sufficient evidence to justify that the present method of enforcement will be ineffective.

9. The Requirement for Lock-Down Would Create A Form of Ubiquitous Law Enforcement, Presenting Civil Liberty Issues

FCC has recently closed field-offices and otherwise reduced the available resources in the field for enforcement against rule-violators. Such enforcement is expensive, requiring specialized equipment, highly-trained staff, and the cooperation of law enforcement such as the U.S. Marshall.

The current rule-making text appears to be an attempt to render such costly enforcement unnecessary through the use of a “digital policeman” in each WiFi device produced, which would ubiquitously prevent law violation before the offense. Such devices effectively lock up their own users, in that the digital device must approve actions of its owner before they are allowed to take place. The advent of a technical capability for *ubiquitous law enforcement* is attractive to regulators and to other parties, for example copyright holders who feel themselves to be overly infringed by the general public have mandated the lock-down of encryption facilities within media devices through the Digital Millenium Copyright Act.

Ubiquitous law enforcement through mandated controls embedded in personal electronic devices presents significant civil liberty issues. Credible social-technological commentators[10] have proposed that the implementation of such controls, potentially starting with enforcement mechanisms regarding relatively trivial law violations like that embedded in DMCA or proposed in the current rule-making text, could create a slippery slope of implementation of controls that might eventually lead to totalitarianism. FCC should not embark upon that slope.

The commenter strongly advises FCC not to turn to embedded enforcement mechanisms as a substitute for finding and citing violators as FCC does today. Certainly the relatively minor offenses theorized in the rule-making text do not mandate it. The potential for negative impact on civil liberty is dire.

[10] Since Orwell. Vernor Vinge has written about the possibilities of more modern equipment. See http://www.code-is-law.org/preface_excerpt.html and http://davidbrin.blogspot.com/2015/01/omniveillance-and-ubiquitous-law.html . Obviously Orwell would have had a lot more to work with in an age in which most people carry microphones, cameras, and radiolocation transmitters with them at all times, and embedded computers are capable of imposing arbitrary rules upon their users, as is the case with our smartphones.

10. Harm to Amateur Radio and to Experimentation

Licensed Radio Amateurs frequently modify WiFi equipment for operation within Part 97. This operation may take place on the frequencies originally intended for the devices, where those frequencies are shared with Amateur Radio. In other cases, transverter systems are used in which WiFi devices work on an intermediate frequency and the communication takes place on a different frequency.

The proceeding as written would lock down WiFi firmware, preventing modification, and prevent the availability of documentation for WiFi integrated circuits that could be used by Radio Amateurs to operate using WiFi devices.

Amateur Radio has historically been a source of innovation for the radio industry and continues to be one today. For example, the *Codec2* software for ultra-low-bandwidth digital voice communication was pioneered by Radio Amateurs for their own use on the air, and is now finding many other uses within industry.

Prevention of the modification of WiFi firmware for other than the intended Part 15 operation will stifle innovation based on WiFi hardware by Radio Amateurs who operate within Part 97 and other experimenters who operate within the present Part 15 or Part 5 regulations. These innovators are very active members of the Open Source developer community today. A rule-making which ends their activities would be contrary to the public interest.

11. Resolving The Conflicts: Crafting Rules that Allow Open Source

To go forward with a regulation that protects the licensed band users from interference and allows Open Source implementations to exist, we must give Open Source developers the same authority and responsibility as manufacturers. We will need to support three kinds of Open Source development of software that governs the on-air operation of WiFi devices:

1. Software development and testing by individuals and small teams under Part 15.
2. Large-scale distribution of Open Source software to "RF-naïve" users who are ignorant of the potential for interference and means of mitigating it.
3. Use under Part 5 or part 97, by experimenters and Radio Amateurs.

Software development concerning WiFi will come in two flavors: modification to the non-WiFi portions of operating systems software that contains WiFi software, and modification of the WiFi software itself.

It is essential that modification to operating systems software that merely *contains* WiFi software proceed without impediment, lest we entirely stop the development of Open Source operating systems like Linux and BSD.

Software development of WiFi device drivers and firmware is carried out by individuals and small teams, and can be managed with the existing methods of enforcement: direction finding and citation of the out-of-specification operator. **The commenter proposes no restriction regarding development of software for WiFi systems, and testing of that software,** due to the fact that this development and testing is carried out by relatively few people at any time, and their motivation is to produce rule-compliant systems can eventually be distributed to the multitudes. Open Source teams are concerned with producing correct software, and thus can be expected to be very cooperative if enforcement ever becomes necessary.

Similarly, **operation under Part 5 or Part 97 rather than Part 15 requires no new restrictions, since enforcement of those services is already well-defined. Modification of WiFi firmware for use under those parts should not be restricted.**

This brings us to the real problem: distribution of software by developers to RF-naïve users, for operation on potentially Millions of devices. In both the case of equipment manufacturers creating original firmware and Open Source developers creating firmware, when that firmware will be packaged with devices or promoted for installation in binary form by large numbers of RF-naïve persons, **FCC should require that the source-code form of the WiFi drivers be approved by a software developer holding the General Radio Operators License with Ship Radar Endorsement (GROL+Radar), and that source code should include the statement of approval and contact information for that developer.**

The GROL+Radar requires a level of RF and Radar expertise on top of the capabilities that are already required of a software developer, and is presently required of the maintainers of radar systems. It's not a perfect fit to the problem at hand, but it's the best license available for the task at the moment. It would be possible, if desired, to craft a GROL endorsement that is specific to WiFi engineering. Material on the GROL+Radar is readily available and the examination is well within the capability of many Open Source developers.

In the case that source code is not made publicly available, for example in commercial WiFi devices that are not Open Source, the same rule is proposed, except that the identification and contact information for the GROL+Radar-licensed developer and his/her employer should be provided electronically with the device in a form that is easily readable by the user, or in documentation accompanying the device. This is compatible with the e-Labeling initiative.

By requiring approval of the *source-code* form of the device drivers rather than the binary form, the attention of the GROL+Radar-licensed developer will be limited to modifications to the WiFi code. Other software developers without licensing can configure, compile, and package the software and distribute it for use.

This proposed process inserts responsible, qualified, human beings into the process in a way that allows certification of WiFi software by either the initial manufacturer or Open Source developers, allows responsible modification of the software in WiFi devices, and removes an unnecessary, unfair, and unworkable portion of the present rule-making text which would work as a ban on Open Source software for use in the control of WiFi devices.

12. The Rationale for Allocation of Additional Spectrum to WiFi At This Time is Questionable, And Does Not Maximize Effective Spectrum Use As Proposed

The rationale for allocating additional spectrum to WiFi is the failure of existing WiFi devices to accommodate the large numbers of devices with which they are now commonly faced. The experience of the *Bufferbloat Project* is that the existing management of Internet packet queueing was defective and severely impaired home networks from making full use of the hardware. Similarly, the reuse of WiFi spectrum is poorly tuned in many current WiFi devices, and they often fail for reasons not connected with the availability of spectrum.

Many WiFi access points are not capable of dispensing a sufficient quantity of network addresses, and do not expire allocated addresses sufficiently quickly to allow their efficient reuse. In many such devices, software runs out of memory, and in general other resources than spectrum. This is due to a lack of hardware capability, misconfiguration, or age.

Many WiFi access points in operation today were not developed with the anticipation that each user would carry as many devices as they now do. Devices that fail for reasons other than spectrum availability often exhibit a common failure mode: it's necessary to unplug them and cold-start them in order to restore network functionality. This is *not* a spectrum problem. It is, however, a behavior that is very familiar to travelers and to the staff of hotels and presentation venues.

The “cellular” paradigm of spectrum reuse, in which the distance between access points is used to provide for reuse of WiFi channels within a facility, is poorly implemented in the WiFi equipment that is commonly deployed in environments where a large number of users are expected, for example hotels and presentation venues. Such implementation does not in general follow any properly-engineered standard. The implementors are often the hotel or venue maintenance staff or relatively unskilled contractors, and rarely have the engineering background necessary to properly design a cellular system of spectrum reuse without using a process pre-engineered for deployment by non-engineers. No such process is available as an open standard today, and thus optimal deployment of WiFi in hotels and presentation venues is rare.

Proper use of the cellular paradigm would require the installation of more wireless access points, but this would be necessary in any case because the 5 GHz spectrum considered for allocation in this rule-making is not usable for WiFi links over distances greater than 100 feet in typical operating scenarios[11].

[11] “Typical operating scenarios” means conventional wireless clients with omnidirectional antennas and conventional access points with omnidirectional or directional antennas, rather than point-to-point links with high-gain antennas. It is possible to achieve very long range point-to-point links using parabolic antennas, elevation of the stations to avoid the problem of Fresnel zone attenuation, and an exclusively line-of-sight signal path. The figures claimed in long-distance WiFi tests can deceive the uninformed that 5 GHz WiFi at the proposed power levels is capable of communication over significant distances in real life. But in general it will be limited to distances of less than 100 feet.

Indeed, the typical distance achieved for the highest data rates is no more than 30 feet.

Thus, this rule-making may be an attempt to throw valuable radio spectrum at a non-spectrum problem. Without additional standards for the configuration and physical deployment of wireless systems in high-number-of-devices environments, that spectrum will likely be wasted. In order to maximize spectrum reuse, the Commission should require manufacturers to develop and promulgate standards for systems deployment in hotels and presentation venues before it considers the allocation of additional frequencies. The Commission should then require the development, distribution, and use of such standards in connection with any such future allocation.